ANNEE SCOLAIRE 2024/2025 Devoir maison $n°1$ A rendre le 04/11/2024	Lycée Jean-Jacques ROUSSEAU Sarcelles

1 Informatique

1.1 Dichotomie

1. Décrire le principe de la recherche d'une solution par dichotomie.
2. On pose $\begin{array}{cccc} f & : & \mathbb{R}\backslash\{-1\} & \to & \mathbb{R} \\ & & x & \mapsto & \frac{x^2-x+3}{x+1} \end{array}$. Écrire une fonction python f qui prend en argument x et qui renvoie la valeur $f(x)$ et les lignes de code permettant de tracer la courbe de f sur $[-0.5, 8]$ ainsi que la droite d'équation $y = 5$ sur le même graphe.
3. Déterminer le nombre de solution de l'équation $f(x) = 5$.
4. Recopier et compléter la fonction python Dichotomie qui se trouve en annexe et qui prend en arguments a, b (où $-1 < a < b$) et ϵ et qui renvoie une approximation d'une solution de l'équation $f(x) = 5$ à ϵ si cette solution est dans $[a, b]$. (On supposera la fonction f écrite à la question 2 déjà implémenté en Python.)

1.2 Une suite récurrente

On pose la fonction $\begin{array}{ccccc} f & : & \mathbb{R}_+^* & \to & \mathbb{R} \\ & & x & \mapsto & \frac{x+2}{x} \end{array}$ et on considère la suite définie par $u_0 \in \mathbb{R}_+^*$ et pour tout $n \in \mathbb{N}$, $u_{n+1} = f(u_n)$.

1. Justifier que l'équation $f(x) = x$ admet exactement une solution α sur $\mathbb{R}_+^*$ et la calculer.
2. Donner les lignes de code python permettant de tracer sur le même graphique la courbe de f et de la droite d'équation $y = x$ sur $[0.01, 3]$.

On se propose à présent d'illustrer informatiquement la progression de la suite pour un u_0 donné. Pour cela on va relier le point (u_0, u_0) à (u_0, u_1) puis à (u_1, u_1) puis à (u_1, u_2) et ainsi de suite. Par exemple pour $u_0 = 3$, on trouve le graphique associé en annexe.

3. Écrire la liste des abscisses et la liste des ordonnées permettant de tracer les segments décrit ci-dessus jusqu'au point de coordonnée (u_5, u_6).
4. Écrire une fonction python Suite prenant en arguments n et u_0 et qui renvoie u_n.
5. Recopier et compléter la fonction ProgressionSuite qui se trouve en annexe et qui prend en argument n et u_0 et qui permet de tracer le dessin rechercher jusqu'au point $(u_n, f(u_n))$. (On suppose les codes précédents déjà implémenté dans Python.)

2 Espace vectoriel

Soit $n \in \mathbb{N}^*$. Soit $A \in \mathcal{M}_n(\mathbb{R})$ et pour tout $\lambda \in \mathbb{R}$ on pose

$$E_\lambda = \{X \in \mathcal{M}_{n,1}(\mathbb{R}) \mid AX = \lambda X\}.$$

1. Montrer que E_λ est un espace vectoriel réel.

On pose à présent $A = \begin{pmatrix} 1 & -3 & 1 & -5 \\ 2 & 0 & -2 & 2 \\ -3 & -3 & 5 & -9 \\ -2 & 0 & 2 & -2 \end{pmatrix}$.

2. Donner les lignes de code Python permettant de calculer $A^4 - 4A^3 + 4A^2$.

Les lignes de codes précédentes renvoie la matrice 0_4.

3. Factoriser le polynôme $x^4 - 4x^3 + 4x^2$.
4. On note λ_0 et λ_1 les deux racines du polynôme précédent (on suppose que $\lambda_0 < \lambda_1$).
   (a) Déterminer une base $\mathcal{B}_0$ de E_{λ_0}.
   (b) Déterminer une base $\mathcal{B}_1$ de E_{λ_1}.
   (c) La famille $\mathcal{B}_0 \cup \mathcal{B}_1$ est-elle une base de $\mathcal{M}_{4,1}(\mathbb{R})$?

3 Intégrales

Les deux questions sont indépendantes :

1. Étude de $\int_0^{+\infty} \frac{t-1}{t^3+1} dt$.
   (a) Factoriser le polynôme $t^3 + 1$.
   (b) Déterminer a, b et c trois réels tels que pour tout $t \in \mathbb{R}_+$,
   $$\frac{t-1}{t^3+1} = \frac{at+b}{t^2-t+1} + \frac{c}{t+1}.$$
   (c) En déduire la nature de l'intégrale $\int_0^{+\infty} \frac{t-1}{t^3+1} dt$ et donner sa valeur dans le cas de convergence.
2. Étude de $\int_{-\infty}^{+\infty} t^2 e^{-t}\, dt$.
   (a) Étudier l'intégrale impropre $\int_0^{+\infty} t^2 e^{-t} dt$.
   (b) Étudier l'intégrale impropre $\int_{-\infty}^{0} t^2 e^{-t} dt$.
   (c) Conclure quand à la nature de $\int_{-\infty}^{+\infty} t^2 e^{-t}\, dt$ et donner sa valeur dans le cas de convergence.

4 Polynômes et séries numériques

On considère la famille de polynôme $\mathcal{B} = \{P_0, P_1, P_2, P_3\}$ définie par :

— $P_0(x) = (x+1)(x+2)(x+3)$
— $P_1(x) = x(x+2)(x+3)$
— $P_2(x) = x(x+1)(x+3)$
— $P_3(x) = x(x+1)(x+2)$

1. Montrer que la famille $\mathcal{B}$ est une famille libre de $\mathbb{R}_3[x]$. (On pourra évaluer une équation vectorielle en certains valeurs de x).
2. En déduire que $\mathcal{B}$ est une base de $\mathbb{R}_3[x]$.
3. Exprimer $3x^2 + 7x + 6$ dans la base $\mathcal{B}$.

On considère la suite $(u_n)_{n\in\mathbb{N}^*}$ où pour tout $n \in \mathbb{N}^*$, $u_n = \frac{3n^2+7n+6}{n(n+1)(n+2)(n+3)}$

4. Justifier à l'aide d'un théorème de comparaison que la série de terme général u_n est convergente.
5. Déterminer quatre réels a, b, c et d tel que pour tout $n \in \mathbb{N}^*$,
$$u_n = \frac{a}{n} + \frac{b}{n+1} + \frac{c}{n+2} + \frac{d}{n+3}.$$
6. En déduire la valeur exacte de $\sum\limits_{n=1}^{+\infty} u_n$.

5 Problème : Série de Bertrand

L'objectif de ce problème est de démontrer le résultat suivant :

> **Théorème :** Soit $(\alpha, \beta) \in \mathbb{R}^2$. La série numérique $\sum\limits_{n\geq 2} \frac{1}{n^\alpha (\ln(n))^\beta}$, dites série de Bertrand, est convergente si et seulement si
> $$\alpha > 1 \ \text{ ou } \ (\alpha = 1 \ \text{ et } \ \beta > 1)$$

On note pour tout $n \in \mathbb{N}\backslash\{0;1\}$ $u_n = \frac{1}{n^\alpha (\ln(n))^\beta}$.

1. Cas $\alpha < 0$: Justifier que dans ce cas la série diverge grossièrement.
2. Cas $\alpha \in [0, 1[$: Justifier que $\frac{1}{n} = \underset{n\to+\infty}{o}(u_n)$ puis conclure dans ce cas.
3. Cas $\alpha > 1$:
   (a) Déterminer une suite $(v_n)_{n\in\mathbb{N}^*}$ tel que $u_n = \underset{n\to+\infty}{o}(v_n)$. (On pourra faire intervenir une constante $\delta \in]1, \alpha[$.)
   (b) Conclure dans ce cas.
4. Cas $\alpha = 1$ et $\beta \leq 0$: Justifier par comparaison avec une série bien connue la divergence de la série de Bertrand dans ce cas.
5. Cas $\alpha = 1$ et $\beta > 0$
   (a) On pose $\begin{array}{cccc} f & : & [2, +\infty[& \to & \mathbb{R} \\ & & x & \mapsto & \frac{1}{x(\ln(x))^\beta} \end{array}$. Dresser le tableau de variation complet de f sur $[2, +\infty[$.

(b) Montrer que pour tout $n \in \mathbb{N} \backslash \{0, 1\}$, on a

$$u_{n+1} \leq \int_n^{n+1} f(t)\, dt \leq u_n$$

(c) En déduire que pour tout $N \in \mathbb{N} \backslash \{0, 1\}$, on a

$$\sum_{n=3}^{N+1} u_n \leq \int_2^{N+1} f(t)\, dt \leq \sum_{n=2}^{N} u_n$$

(d) En déduire que la série de Bertrand $\sum\limits_{n=2}^{\infty} u_n$ et l'intégrale impropre $\int_2^{\infty} f(t)\, dt$ ont la même nature.

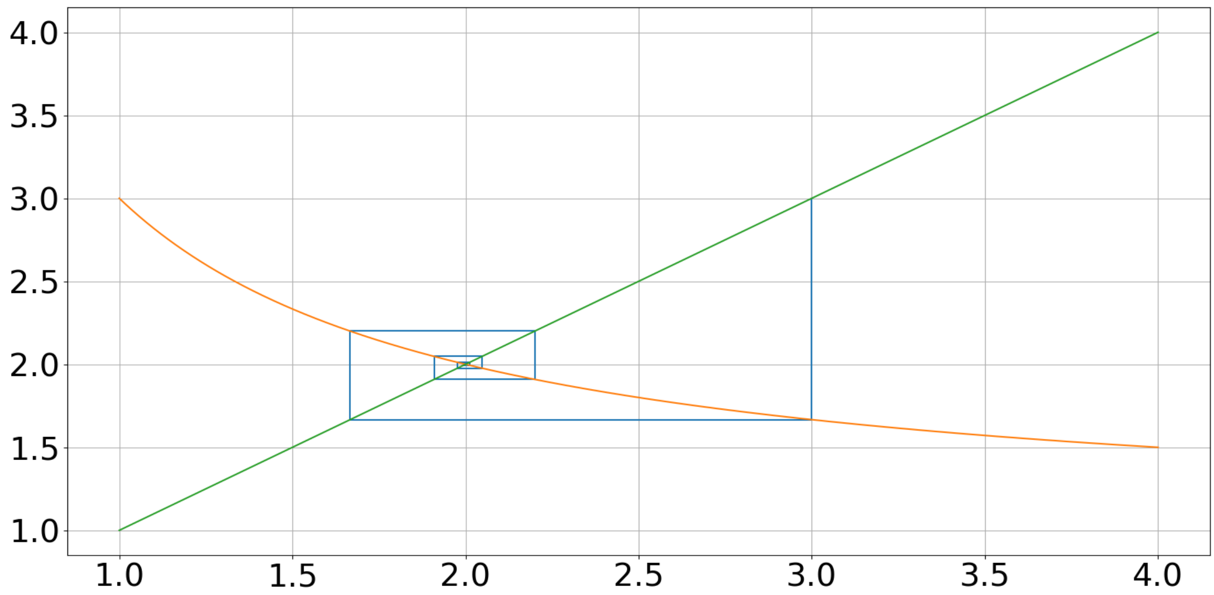
(e) A l'aide du changement de variable $x = \ln(t)$ calculer pour tout $(a, b) \in [2, +\infty[^2$, $\int_a^b f(t)\, dt$. (On pourra séparer le cas $\beta = 1$ du cas $\beta \neq 1$.)

(f) En déduire pour quelles valeurs de β l'intégrale impropre $\int_2^{\infty} f(t)\, dt$ est convergente.

(g) Conclure quand à la série de Bertrand.

6. Démontrer le théorème.

Annexe




```
def Dichotomie(a,b,Eps) :
        if (f(a)-5)*(f(b)-5) > 0 : # Que se passe t il dans ce cas ?
                return · · · · · · · · ·
        elif f(a) > 5 :
                c = (a+b)/2
                while · · · · · · · · ·  :
                        if · · · · · · · · ·  :
                                a = · · ·
                                c = · · ·
                        else :
                                b = · · ·
                                c = · · ·
                return c
        else :
                c = (a+b)/2
                while · · · · · · · · ·  :
                        if · · · · · · · · ·  :
                                a = · · ·
                                c = · · ·
                        else :
                                b = · · ·
                                c = · · ·
                return c
```

```
def ProgressionSuite(n,u) :
    U = ········· # Liste de termes de la suite u
    XP = []
    YP = []
    for k in range(n) : # On procède par couple de couple de coordonnées
        XP = ·········
        YP = ·········
    plt.plot(XP,YP)
    X = ········· # On trace ici la courbe de f ainsi que la droite d'équation y=x
    Y = ·········
    ·········
    ·········
    plt.grid()
    plt.show()
```